Non-Parametric Contextual Stochastic Search

Abbas Abdolmaleki[1,2,3], Nuno Lau[1], Luis Paulo Reis[2,3], Gerhard Neumann[4]

***Abstract*— Stochastic search algorithms are black-box optimizer of an objective function. They have recently gained a lot of attention in operations research, machine learning and policy search of robot motor skills due to their ease of use and their generality. Yet, many stochastic search algorithms require relearning if the task or objective function changes slightly to adapt the solution to the new situation or the new context. In this paper, we consider the contextual stochastic search setup. Here, we want to find multiple good parameter vectors for multiple related tasks, where each task is described by a continuous context vector. Hence, the objective function might change slightly for each parameter vector evaluation of a task or context. Contextual algorithms have been investigated in the field of policy search, however, the search distribution typically uses a parametric model that is linear in the some hand-defined context features. Finding good context features is a challenging task, and hence, non-parametric methods are often preferred over their parametric counter-parts. In this paper, we propose a non-parametric contextual stochastic search algorithm that can learn a non-parametric search distribution for multiple tasks simultaneously. In difference to existing methods, our method can also learn a context dependent covariance matrix that guides the exploration of the search process. We illustrate its performance on several non-linear contextual tasks.**

I. Introduction

Stochastic search algorithms are gradient-free black-box optimizers of some objective function dependent on a high dimensional parameter vector. These algorithms only make weak assumption on the structure of underlying objective function. They only use the objective function values of the parameters that we want to optimise and don't require gradients or higher derivatives of the objective function. For example, in robotics, we can directly evaluate the objective function value for a parameter vector of a controller by executing that parameter vector and using the return of an episode. Stochastic search algorithms [1], [2], [3] typically maintain a search distribution over the parameters that we want to optimise. This search distribution is used to create samples of the parameter vector. Subsequently, the performance of the sampled parameters is evaluated. Using the samples and their evaluations, a new search distribution is computed by either computing gradient based updates [2], [3], evolutionary strategies [1], the cross-entropy method [4], path integrals [5], or information-theoretic policy updates [6], [7]. However, many of the mentioned algorithms can not be applied for multi-task learning. Therefore, if the task setup or objective function changes slightly, relearning is needed to adapt the solution to the new situation or the new context. For example, consider optimising the parameters of a humanoid soccer robot controller to kick a ball. Once the characteristics of the ball, such as weight of ball, or objective function, such as the desired kick distance changes, relearning of the desired kicking motion is needed. Therefore we would like to learn a context-dependent function that generates optimal parameters for a desired task or context. Contextual search algorithms such as contextual REPS [6] have been investigated in the field of policy search. These algorithms maintain a parametric context-dependent function as the mean of a Gaussian policy which is linear in the some hand-defined context features. Firstly finding good context features to capture the non-linearity of the desired context-dependent function is a challenging task, and hence, non-parametric methods are often preferred over their parametric counter-parts. Secondly only the mean of the Gaussian search distribution is context-dependent while the covariance matrix is fixed for all contexts. Hence, these algorithms find a covariance matrix that, in average, is good for all the contexts. However, in order to guide the policy search it is desirable to have a fully context-dependent search distribution with optimal mean and covariance matrix for a specific context. Therefore, we introduce a non-parametric contextual policy search method that can learn non-linear context-dependent functions and leverage from a fully context-dependent search distribution. We name our method local Covariance Estimation with Controlled Entropy Reduction (local CECER). We will show that local CECER performs favourably.

A. Problem Statement

Given a query context vector $\boldsymbol{s}^*$ with m dimensions which defines a task, we want to find a non parametric context-dependent function $m_*(\boldsymbol{s}) : \mathbb{R}^m \rightarrow \mathbb{R}^n$ that generates a parameter vector $\boldsymbol{\theta}^*$ with n dimensions such that it maximizes an objective function $R(\boldsymbol{\theta}, \boldsymbol{s}) : \{\mathbb{R}^n, \mathbb{R}^m\} \rightarrow \mathbb{R}$. The only accessible information on $R(\boldsymbol{\theta}, \boldsymbol{s})$ are evaluations $\{R^{[k]}\}_{k=1...N}$ of samples $\{\boldsymbol{s}^{[k]}, \boldsymbol{\theta}^{[k]}\}_{k=1...N}$, where k is the index of the sample and N is number of samples. Essentially the goal of local CECER is to generate a dataset $\{\boldsymbol{s}^{[k]}, \boldsymbol{\theta}^{[k]}\}_{k=1...N}$ that contains the optimal parameters for the corresponding context vectors. With this data set, the optimal vector $\boldsymbol{\theta}^*$ for a given context $\boldsymbol{s}^*$ can be found in a non-parametric fashion using locally weighted linear regression method.

B. Related Work

In order to generalize a parameter vector to the other contexts, for example, kicking the ball for different distances,

[1]DETI/IEETA, University of Aveiro, Aveiro, Portugal
[2]LIACC, University of Porto, Porto, Portugal
[3]DSI, University of Minho, Braga, Portugal
[4]CLAS, IAS, TU Darmstadt, Darmstadt, Germany
{abbas.a,nunolau}@ua.pt, lpreis@dsi.uminho.pt, geri@robot-learning.de

typically the parameters are optimized for several target contexts independently. Subsequently, regression methods are used to generalize the optimized contexts to a new, unseen context. Although such approaches have been used successfully, they are time consuming and inefficient in terms of the number of needed training samples as optimizing for different contexts and the generalization between optimized parameters for different contexts are two independent processes[8]. Hence, it is desirable to learn the selection of the parameter for multiple tasks at once without restarting the learning process once we see a new task. This problem setup is also known as contextual policy search [6], [9]. Such a multi-task learning capability was established for information-theoretic policy search algorithms [10], such as the Contextual Relative Entropy Policy Search (CREPS) algorithm [6]. Contextual REPS was originally applicable for the problems with linear generalization over contexts or tasks. In [11], contextual REPS was extended for tasks with non-linear generalization over contexts, by using radial basis functions resulting in contextual RBF-REPS. However due to use of radial basis functions, this method can suffer from curse of dimensionality and also finding a good settings for RBFs is a challenging task. Moreover, the update rule of the search distribution in REPS and RBF-REPS is not fully context-dependent. In addition, REPS and RBF-REPS can suffer from premature convergence. In [12] Covariance Estimation with Controlled Entropy Reduction (CECER) algorithm was introduced to alleviate the premature convergence problem of REPS. Our new algorithm local CECER leverage from both nonlinear generalization over contexts and fully context dependent search distribution update rule while it also uses CECER algorithm concept[12] to avoid premature convergence.

II. Non-Parametric Contextual Stochastic Search

local CECER is a non-parametric policy search approach. Therefore we always maintain a dataset $\mathcal{D} = \{\boldsymbol{s}^{[k]}, \boldsymbol{\theta}^{[k]}, \boldsymbol{\Sigma}^{[k]}\}_{k=1\dots N}$ with N samples that contains the contexts, parameters pair $\{\boldsymbol{s}^{[k]}, \boldsymbol{\theta}^{[k]}\}$ and its evaluation $R^{[k]}$ as well as the covariance matrix $\boldsymbol{\Sigma}^{[k]}$ that has been used to generate parameters $\boldsymbol{\theta}^{[k]}$. In each iteration, given a new query context $\boldsymbol{s}^*$, we first compute a locality (similarity) weighting $w^{[k]}$ for each sample with respect to the query context $\boldsymbol{s}^*$. We use these locality weightings to compute a weight or pseudo probability $d^{[k]}$ for each sample in the data set and subsequently, we obtain a local Gaussian search distribution $\pi_*(\boldsymbol{\theta}|\boldsymbol{s})$. We use the search distribution $\pi_*(\boldsymbol{\theta}|\boldsymbol{s}^*)$ to create a sample $\boldsymbol{\theta}^*$ for the query context $\boldsymbol{s}^*$. Subsequently, the evaluation R^* of $\{\boldsymbol{s}^*, \boldsymbol{\theta}^*\}$ is obtained by querying the objective function $R(\boldsymbol{\theta}^*, \boldsymbol{s}^*)$. Afterwards, we update the dataset with the new sample $\{\boldsymbol{s}^*, \boldsymbol{\theta}^*, \boldsymbol{\Sigma}^*, R^*\}$ [1]. We also use the locality weightings to update the covariance matrices of neighboured samples of query context $\boldsymbol{s}^*$ to improve the estimate of their local covariance matrix. This process will run iteratively until a stopping criteria is met. We start by explaining how the weights or pseudo probabilities $d^{[k]}$ are computed and, after that, we explain the local Gaussian search distribution update rules.

[1]Please note that the way we sample contexts $\boldsymbol{s}^{[k]}$ depends on the task. Throughout this paper we use a uniform distribution to sample contexts $\boldsymbol{s}$.

Algorithm 1 local CECER Weights Computation

Input : Data Set $\mathcal{D}\{\boldsymbol{s}^{[k]}, \boldsymbol{\theta}^{[k]}, R^{[k]}, \boldsymbol{\Sigma}^{[k]}\}_{k=1\dots N}$, the query context $\boldsymbol{s}^*$

Compute the locality weightings $w^{[k]}$ for each sample:

$$w^{[k]} = \exp(-0.5|\boldsymbol{s}^{[k]} - \boldsymbol{s}^*|^2/b)\ , Z_w = \sum_{k=1}^{N} w^{[k]}.$$

Compute the weights $d^{[k]}$ for each sample:

1- Optimize the dual function g for η and $\boldsymbol{w}$

$$\begin{aligned} g(\eta, \boldsymbol{w}) =& \eta\epsilon + \hat{\boldsymbol{\phi}}^T \boldsymbol{w} \\ &+ \eta \log\left(\sum_{K=1}^{N} \frac{w^{[k]}}{Z_w} \exp\left(\frac{R^{[k]} - \boldsymbol{\phi}(\boldsymbol{s}^{[k]})^T \boldsymbol{w}}{\eta}\right)\right) \\ \hat{\boldsymbol{\phi}} =& \sum_{k=1}^{N} \frac{w^{[k]}}{Z_w} \boldsymbol{\phi}(\boldsymbol{s}^{[k]}). \end{aligned}$$

2- Compute weights

$$d^{[k]} = \frac{w^{[k]} \exp\left(\frac{R^{[k]} - \boldsymbol{\phi}(\boldsymbol{s}^{[k]})^T \boldsymbol{w}}{\eta}\right)}{Z}\ ,\ Z = \sum_{k=1}^{N} d^{[k]}.$$

A. Weight Computation

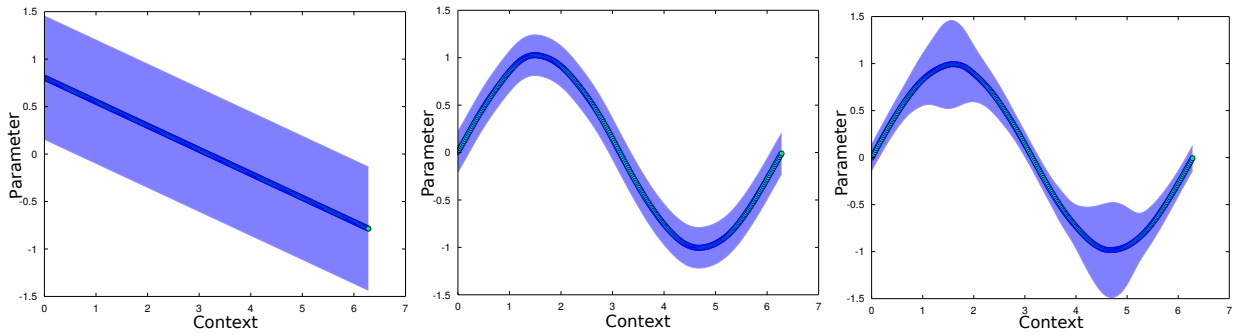
Given query context $\boldsymbol{s}^*$, local CECER first computes a locality weighting $w^{[k]}$ for each sample. We use a normalized squared exponential kernel i.e.,

$$w(\boldsymbol{s}) = \frac{k(\boldsymbol{s}, \boldsymbol{s}^*)}{\int k(\boldsymbol{s}, \boldsymbol{s}^*) d\boldsymbol{s}}\ ,\ k(s, s^*) = \exp(-0.5|\boldsymbol{s} - \boldsymbol{s}^*|^2/b).$$

Now we use this locality weightings to obtain a pseudo probability or a weight for each sample in our data set. To do so we find the joint probabilities $p_*(\boldsymbol{s}, \boldsymbol{\theta}) = \pi_*(\boldsymbol{\theta}|\boldsymbol{s})\mu_*(\boldsymbol{s})$ by optimizing the following performance criteria[6] for each new query context $\boldsymbol{s}^*$, i.e.,

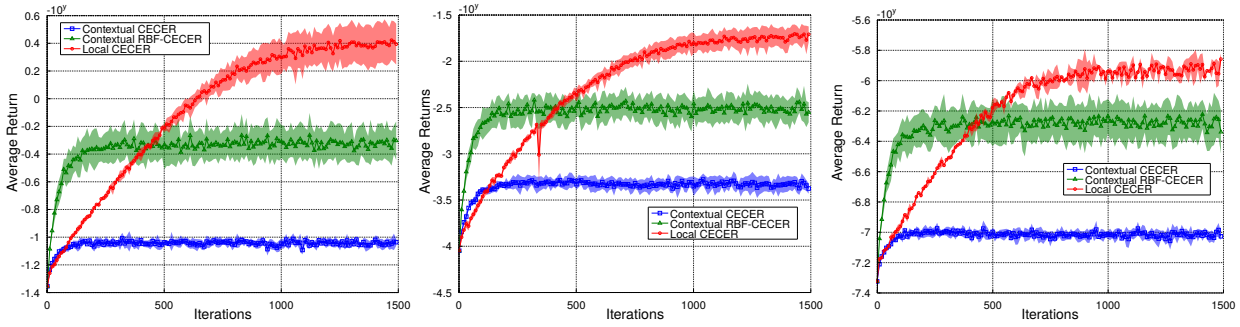
$$\begin{aligned} &\max_{p_*} \iint p_*(\boldsymbol{s}, \boldsymbol{\theta}) \mathcal{R}_{\boldsymbol{s}\boldsymbol{\theta}} d\boldsymbol{s} d\boldsymbol{\theta} \\ &\text{s.t. } \epsilon \geq \text{KL}(p_*(\boldsymbol{s}, \boldsymbol{\theta}) || \mu_*(\boldsymbol{s}) q(\boldsymbol{\theta}|\boldsymbol{s})), \\ &\hat{\boldsymbol{\phi}} = \iint p_*(\boldsymbol{s}, \boldsymbol{\theta}) \boldsymbol{\phi}(\boldsymbol{s}) d\boldsymbol{s} d\boldsymbol{\theta}, \\ &1 = \iint p_*(\boldsymbol{s}, \boldsymbol{\theta}) d\boldsymbol{s} d\boldsymbol{\theta}. \end{aligned} \tag{1}$$

The key idea behind this optimization program is to ensure a smooth and stable learning process by bounding the Kullback-Leibler divergence between the old local search distribution and the newly estimated local search distribution while maximising the expected return for the given context $\boldsymbol{s}^*$. Where $\mathcal{R}_{\boldsymbol{s}\boldsymbol{\theta}}$ denotes the expected performance when



(a) Contextual CECER (b) Contextual RBF-CECER (c) Local CECER

Fig. 1. The learned policy by CECER, RBF-CECER and local CECER for sin task. Darker blue shows the mean of the search distribution for each context. While the shaded area with lighter blue shows the variance of the search distribution around the mean for each context. The results show that Local CECER and RBf-CECER can learn non-linear policies. Moreover local CECER is able to learn a search distribution that both the mean and variance of the distribution is context dependent which is a desirable feature. As you can see in CECER and RBF-CECER the variance of the search distribution for all the contexts is fixed.



(a) Sphere (b) Rosenbrock (c) Cigar

Fig. 2. The performance comparison of stochastic search methods for optimising contextual version of standard functions (a)Sphere, (b)Rosenbrock and (c)Cigar, The results show that local CECER outperforms both CECER and RBF-CECER.

evaluating parameter vector $\boldsymbol{\theta}$ in context $\boldsymbol{s}$, q is the old sample distribution. While $\mu_(\boldsymbol{s})$ is the context distribution, $\mu_*(\boldsymbol{s})$ denotes the local context distribution with respect to context $\boldsymbol{s}^*$ which can be obtained using the locality weighting function $w(\boldsymbol{s})$ i.e.,

$$\mu_*(\boldsymbol{s}) = \frac{w(\boldsymbol{s})\mu(s)}{\int \mu(\boldsymbol{s})w(\boldsymbol{s})d\boldsymbol{s}}.$$

In addition $\hat{\boldsymbol{\phi}} = \int_{\boldsymbol{s}} \mu_*(\boldsymbol{s})\boldsymbol{\phi}(\boldsymbol{s})d\boldsymbol{s}$ is the expected feature vector for the local context distribution $\mu_*(\boldsymbol{s})$, a given query context $\boldsymbol{s}^*$ and a given feature space $\boldsymbol{\phi}$. This optimization problem can be solved efficiently by the method of Lagrangian multipliers [13]. The solution for $p_*(\boldsymbol{s}, \boldsymbol{\theta})$ is now given by

$$p_*(\boldsymbol{s}, \boldsymbol{\theta}) \propto q(\boldsymbol{\theta}|\boldsymbol{s})\mu_*(\boldsymbol{s}) \exp\left((\mathcal{R}_{\boldsymbol{s}\boldsymbol{\theta}} - V(\boldsymbol{s}))/\eta\right),$$

where $V(\boldsymbol{s}) = \boldsymbol{\phi}(\boldsymbol{s})^T\boldsymbol{w}$ is a context dependent baseline which is subtracted from the return $\mathcal{R}_{\boldsymbol{s}\boldsymbol{\theta}}$. The parameters $\boldsymbol{w}$ and η are Lagrangian multipliers that can be obtained by optimizing the dual function, given as

$$g(\eta, \boldsymbol{w}) = \eta\epsilon + \hat{\boldsymbol{\phi}}^T\boldsymbol{w} + \eta \log\left(\iint \mu_*(\boldsymbol{s})q(\boldsymbol{\theta}|\boldsymbol{s}) \exp\left(\frac{\mathcal{R}_{\boldsymbol{s}\boldsymbol{\theta}} - \boldsymbol{\phi}(\boldsymbol{s})^T\boldsymbol{w}}{\eta}\right) d\boldsymbol{\theta}d\boldsymbol{s}\right). \quad (2)$$

This policy update results in a weight or pseudo probability

$$d^{[k]} = w^{[k]} \exp\left((R^{[k]} - V(\boldsymbol{s}^{[k]}))/\eta\right)$$

for each sample $[\boldsymbol{s}^{[k]}, \boldsymbol{\theta}^{[k]}]$ given a query context $\boldsymbol{s}^*$ where

$$w^{[k]} = \frac{k(\boldsymbol{s}, \boldsymbol{s}^*)}{Z_w}, Z_w = \sum_{k=1}^{N} w^{[k]}.$$

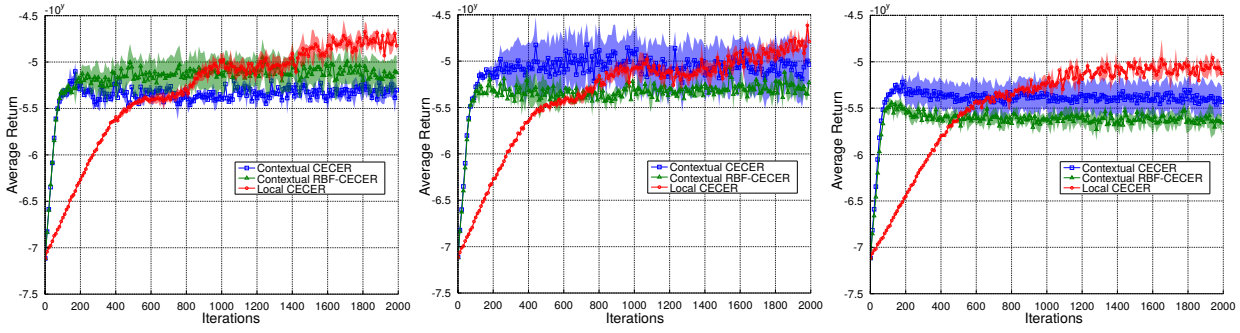
See Algorithm 1 for a compact representation of the weight computation of local CECER algorithm. In the next section we show how we can use these pseudo probabilities to estimate a local Gaussian search distribution $\pi_*(\boldsymbol{\theta}|\boldsymbol{s})$ exclusively for the query context $\boldsymbol{s}^*$.

B. Search Distribution Update Rule

Given dataset $\{\boldsymbol{s}^{[k]}, \boldsymbol{\theta}^{[k]}, w^{[k]}, \boldsymbol{\Sigma}^{[k]}, d^{[k]}\}_{k=1\ldots N}$ and a query context $\boldsymbol{s}^*$, we want to find a local linear Gaussian search distribution

$$\pi_*(\boldsymbol{\theta}|\boldsymbol{s}) = \mathcal{N}\left(\boldsymbol{\theta}|\boldsymbol{m}_{\pi_*}(\boldsymbol{s}) = \boldsymbol{A}_{\pi_*}^T\varphi(\boldsymbol{s}), \boldsymbol{\Sigma}_{\pi_*}\right),$$

by finding $\boldsymbol{A}_{\pi_*}$ and $\boldsymbol{\Sigma}_{\pi_*}$. Where $\varphi(\boldsymbol{s})$ is an arbitrary feature function of context $\boldsymbol{s}$, $\boldsymbol{A}_{\pi_*}^T$ is the gain matrix and $\boldsymbol{\Sigma}_{\pi_*}$ is the covariance matrix. Throughout this paper $\boldsymbol{\varphi}(\boldsymbol{s}) = [1 \;\; \boldsymbol{s}]$,



(a) 1 dim context (b) 2 dim contexts (c) 3 dim contexts

Fig. 3. Performance evaluation on hole reaching task up to 3 dim contextual setup. The results show that local CECER outperforms other algorithms and can learn the task while the other algorithms can not learn the task. Please also see figure 4 and figure 5

which results in linear generalization over contexts. Therefore we need update rules for updating the mean function $\boldsymbol{m}_{\pi_*}$ and for updating the covariance matrix $\boldsymbol{\Sigma}_{\pi_*}$.

1) Context-Dependent Mean-Function: In order to find $\boldsymbol{m}_{\pi_*}$, the parameters $\boldsymbol{A}_{\pi_*}$ can be obtained by the weighted linear ridge regression

$$\boldsymbol{A}_{\pi_*} = \left(\boldsymbol{\Phi}^T \boldsymbol{D} \boldsymbol{\Phi} + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{\Phi}^T \boldsymbol{D} \boldsymbol{U}, \tag{3}$$

where $\boldsymbol{\Phi}^T = [\boldsymbol{\varphi}^{[1]}, ..., \boldsymbol{\varphi}^{[N]}]$ contains the feature vector for all samples, $\boldsymbol{U}^T = [\boldsymbol{\theta}^{[1]}, ..., \boldsymbol{\theta}^{[N]}]$ contains all the sample parameters, $\boldsymbol{D}$ is the diagonal weighting matrix containing the weightings $d^{[k]}$ and $\lambda \boldsymbol{I}$ is a regularization term.

2) Context-Dependent Covariance Matrix: Similar to Standard Contextual REPS we can directly use the weighted sample covariance matrix $\boldsymbol{S}_*$ as local covariance estimate $\boldsymbol{\Sigma}_{\pi_*}$ which is obtained by

$$\boldsymbol{S}_* = \frac{\sum_{k=1}^{N} d^{[k]} \left(\boldsymbol{\theta}^{[k]} - \boldsymbol{A}_{\pi_*}^T \boldsymbol{\varphi}(\boldsymbol{s}^{[k]})\right) \left(\boldsymbol{\theta}^{[k]} - \boldsymbol{A}_{\pi_*}^T \boldsymbol{\varphi}(\boldsymbol{s}^{[k]})\right)^T}{Z}, \tag{4}$$

$$Z = \frac{(\sum_{k=1}^{N} d^{[k]})^2 - \sum_{k=1}^{N} (d^{[k]})^2}{\sum_{k=1}^{N} (d^{[k]})}.$$

However it has been shown that the sample covariance matrix from Equation 4 can cause premature convergence [12]. In order to alleviate this problem, similar to CECER [12] we combine the local old covariance matrix and the local sample covariance matrix from Equation 4, i.e.,

$$\boldsymbol{\Sigma}_{\pi_*} = (\lambda) \boldsymbol{S}_* + (1 - \lambda) \boldsymbol{\Sigma}_{q_*}.$$

In local CECER, the local old covariance matrix also depends on context query $\boldsymbol{s}^*$. Therefore we estimate the local old covariance $\boldsymbol{\Sigma}_{q_*}$ by a weighted average of covariance matrices $\boldsymbol{\Sigma}^{[k]}$ in the dataset. We use the locality weightings $w^{[k]}$ as weights, i.e.,

$$\boldsymbol{\Sigma}_{q_*} = \sum_{k=1}^{N} \frac{w^{[k]}}{Z_w} \boldsymbol{\Sigma}^{[k]}.$$

There are different ways to determine the interpolation factor $\lambda \in [0, 1]$ between the sample covariance matrix $\boldsymbol{S}_*$ and the old covariance matrix $\boldsymbol{\Sigma}_{q_*}$. For example, see the rank-μ update in CMA-ES algorithm [1]. Similar to CECER, the factor $\lambda \in [0, 1]$ is chosen in such a way that the entropy of the new search distribution is reduced by a certain amount ΔH. The entropy of a Gaussian distribution only depends on its covariance $\boldsymbol{\Sigma}_{\pi_*}$ and is given by

$$H(\boldsymbol{\Sigma}_{\pi_*}) = 0.5\big(n + n \log(2\pi) + \log |\boldsymbol{\Sigma}_{\pi_*}|\big).$$

Therefore, λ is chosen such that a desired entropy reduction is achieved, i.e.,

$$H(\boldsymbol{\Sigma}_{q_*}) - H(\lambda \boldsymbol{\Sigma}_{q_*} + (1 - \lambda) \boldsymbol{S}_*) = \Delta H.$$

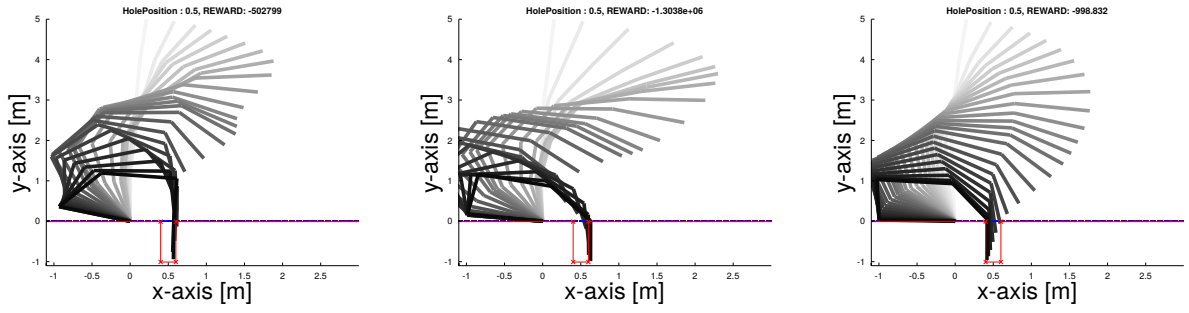
The parameter ΔH is a user-defined parameter to tune the algorithm. After obtaining $\boldsymbol{\Sigma}_{\pi_*}$ we update all the covariance matrices in the dataset using locality weightings i.e.,

$$\boldsymbol{\Sigma}^{[k]} = \beta \boldsymbol{\Sigma}_{\pi_*} + (1 - \beta) \boldsymbol{\Sigma}^{[k]}, \beta = \frac{w^{[k]}}{Z_w}.$$

We subsequently use the policy $\pi_*(\boldsymbol{\theta}|\boldsymbol{s}^*)$ to generate a new parameter $\boldsymbol{\theta}^*$ for the context query $\boldsymbol{s}^*$ and add the new sample $\{\boldsymbol{s}^*, \boldsymbol{\theta}^*, R^*, \boldsymbol{\Sigma}_{\pi_*}\}$ to our dataset. In this paper given that we always want to keep N samples in our dataset, we replace the new sample with the oldest sample if number of samples exceeds N. However other dataset update strategies based on context density could be implemented.

III. Experiments

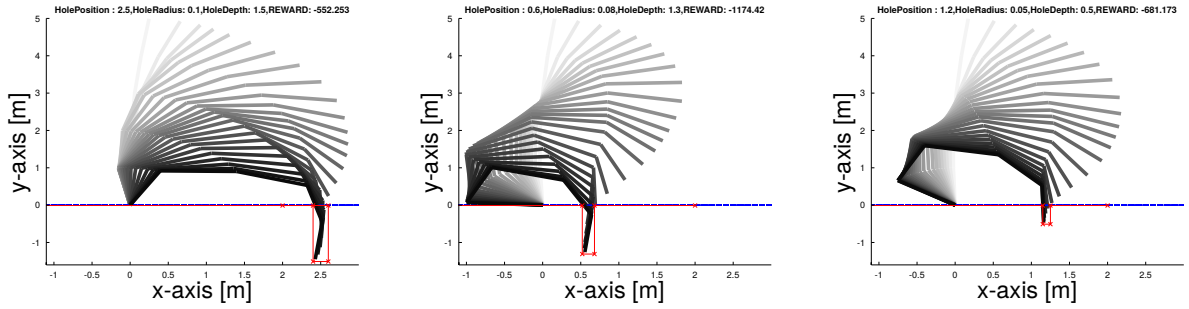
In this section we compare our algorithm local CECER with contextual CECER and contextual RBF-CECER [12] which are improved versions of standard contextual REPS and RBF-REPS[11] respectively. Contextual RBF-CECER is similar to contextual CECER with the difference that RBF-CECER use radial basis functions for non-linear generalization over contexts[11]. We chose three different contextual toy tasks. We use a simple standard $\sin$ function with one parameter to show that local CECER can learn non-linear policies with context-dependent covariance matrix. In the second series, we use standard optimization test functions [14], such as the Sphere, the Rosenbrock and the Cigar function. We extend these functions to be applicable for contextual setting with non-linear generalization over contexts. The task is to find the optimum 15 dimensional parameter vector $\boldsymbol{\theta}$ for a given 2 dimensional context $\boldsymbol{s}$. Furthermore



(a) Contextual CECER (b) Contextual RBF-CECER (c) Local CECER

(d) Contextual CECER (e) Contextual RBF-CECER (f) Local CECER

Fig. 4. A 5-link robot has to reach the bottom of a hole (20 cm wide and 1 m deep) at time step 100 centering at a point varying from 0.5 to 2.5 without any collision with the ground or the hole wall. The red lines show the ground and the hole. The postures of the resulting motion are shown as overlay, where darker postures indicate a posture which is close in time to the bottom of the hole. In the title of each figure, you can see the given context value and gained reward by each algorithm. In this task while local CECER successfully complete the task for both contexts, the other algorithms fail to complete the tasks .(Please see the resulting rewards in the title of the figures)



(a) Local CECER-3 dim contexts (b) Local CECER-3 dim contexts (c) Local CECER-3 dim contexts

Fig. 5. The learned policy by local CECER for 3 dimension contextual hole reaching task. As you can see local CECER could learn the task for 3 dim context while the other algorithms didn't learn a reasonable policy that we could show. You can see the value of query contexts as well as obtained reward in the title of figures.

for the comparisons we use a 5-link planar robot that has to reach the bottom of a given hole without collision with the walls of the hole in task space. We used dynamic movement primitives (DMPs) [15] as underlying policy representation with 30 parameters (five basis functions per dimension and 1 goal position per dimension). For this task, we use three contexts which are the position of the hole, width and depth of the hole. We use hole reaching task with one dimensional context(hole position), two dimensional context (hole position and hole width) and three dimensional context. Figure 4 shows the setup. We show the average as well as two times the standard deviation of the results over 5 trials for each experiment. Note that the y-axis of all plots is in a logarithmic scale.

A. *Sinus Function Task*

In this task, the reward function is given as the distance to a sin function and the distance punishment varies for the context variable (i.e. some contexts are harder to achieve) i.e., $\mathcal{R}_{\boldsymbol{s\theta}} = -(\boldsymbol{\theta} - \sin(\boldsymbol{s}))^2 \times (1 + 5\cos(\boldsymbol{s}))^2$. Both, context and parameter to learn, are 1 dimensional. In Figure 1, we show the mean and variance of the search distribution for each context. Figure 1 shows that RBF-CECER and local CECER both can capture the non-linearity of the function, however only local CECER has different search distribution variance for each context. This experiment shows that only local CECER can learn which context is harder to achieve (less variance) which is easier achieve (high variance).

B. Standard Optimization Test Functions

We chose three standard optimization functions which are the Sphere function $f(\boldsymbol{s}, \theta) = \sum_{i=1}^{p} \boldsymbol{x}_i^2$ and the Rosenbrock function $f(\boldsymbol{s}, \theta) = \sum_{i=1}^{p-1} [100(\boldsymbol{x}_{i+1} - \boldsymbol{x}_i^2)^2 + (1 - \boldsymbol{x}_i)^2]$ and also a function which is known as Cigar function $f(\boldsymbol{s}, \theta) = x_1^2 + 10^6 \sum_{i=2}^{p} \boldsymbol{x}_i^2$. Where $\boldsymbol{x} = \boldsymbol{\theta} + \sin(\boldsymbol{A}\boldsymbol{s})$. The matrix $\boldsymbol{A}$ is a constant matrix that was chosen randomly. In our case, because the context $\boldsymbol{s}$ is 2 dimensional, $\boldsymbol{A}$ is a $n \times 2$ dimensional vector. Now, the optimum $\boldsymbol{\theta}$ for these functions is non-linearly dependent on the given context $\boldsymbol{s}$. The initial search area of $\boldsymbol{\theta}$ for all experiments is restricted to the hypercube $-5 \leq \boldsymbol{\theta}_i \leq 5, i = 1, \ldots, p$ and contexts are samples uniformly from interval $0 \leq \boldsymbol{s}_i \leq 3, i = 1, \ldots, z$ where z is dimension of the context space $\boldsymbol{s}$. In our experiments, the mean of the initial distribution to generate the initial data set have been chosen randomly in the defined search area.

a) Algorithmic Comparison: We generate 2500 samples in the first iteration and in each iteration, we generated 1 new samples and we always keep last 2500 samples. The results in figure 2 shows that local CECER outperforms both contextual CECER and contextual RBF-CECER.

C. Planar Hole Reaching

In this task, we used a 5-link planar robot with DMPs [15] as the underlying control policy. Each link had a length of $1m$. The robot is modelled as a decoupled linear dynamical system. For completing the hole reaching task, the robot end effector has to reach the bottom of a hole with a width varying from 10cm to 40cm, centering at a point varying from 0.5m to 2.5m and with a depth varying from 50cm to 1.5m without any collision with the ground or the hole wall. The reward was given by a quadratic cost term for the desired final point, quadratic costs for high accelerations and quadratic costs for collisions with the environment. Note that this performance function is discontinuous due to the cost for collisions. The DMPs goal attractor for reaching the final state in this task is unknown and need to also be learned. Hence, our parameter vector had 30 dimensions. The learning setup is shown in Figure 4.

b) Algorithmic Comparison: For the planar task we generated 2500 samples in the first iteration and 1 new samples in each iteration. We always keep last 2500 samples. We compare all three algorithms in three different contextual settings up to three dimensional context setting. The results in Figure 3 shows that local CECER outperforms the other two algorithms in all three different contextual settings. Figure 4 and Figure 5 shows the learned policies for 1 dimensional context, which is the hole position, and three dimensional context which are the hole position, the hole width and the hole depth. The results show that local CECER could successfully learn for all the query contexts while the other algorithms failed to learn this task.

IV. Conclusion

Multi task learning is an important feature for a robot learning algorithm as a robot usually needs to quickly adapt to new situations. Therefore, in this paper, we investigated a non-parametric contextual stochastic search method called local CECER. We showed that local CECER leverages from a fully context dependent policy update and it is able to learn non-linear policies. We showed that local CECER outperforms the other contextual algorithms. For the future work we investigate the methods to set the bandwidth of the kernel function automatically.

V. Acknowledgment

The work was partially funded by the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 and by FCT Portuguese Foundation for Science and Technology under projects PEst-OE/EEI/UI0027/2013 and UID/CEC/00127/2013 (IEETA and LIACC). The work was also funded by project EuRoC, reference 608849 from call FP7-2013-NMP-ICT-FOF.

References

[1] N. Hansen, S.D. Muller, and P. Koumoutsakos. Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). *Evolutionary Computation*, 2003.

[2] Y. Sun, D. Wierstra, T. Schaul, and J. Schmidhuber. Efficient Natural Evolution Strategies. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation(GECCO)*, 2009.

[3] T. Rückstieß, M. Felder, and J. Schmidhuber. State-dependent Exploration for Policy Gradient Methods. In *Proceedings of the European Conference on Machine Learning (ECML)*, 2008.

[4] S. Mannor, R. Rubinstein, and Y. Gat. The Cross Entropy method for Fast Policy Search. In *Proceedings of the 20th International Conference on Machine Learning(ICML)*, 2003.

[5] E. Theodorou, J. Buchli, and S. Schaal. A Generalized Path Integral Control Approach to Reinforcement Learning. *The Journal of Machine Learning Research*, 2010.

[6] A. Kupcsik, M. P. Deisenroth, J. Peters, and G. Neumann. Data-Efficient Contextual Policy Search for Robot Movement Skills. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2013.

[7] A. Abdolmaleki, R. Lioutikov, J. Peters, N. Lua, L.P. Reis, and G. Neumann. Model Based Relative Entropy Stochastic Search. In *Advances in Neural Information Processing Systems (NIPS), MIT Press*, 2015.

[8] Bruno Da Silva, George Konidaris, and Andrew Barto. Learning parameterized skills. *International Conference on Machine Learning (ICML)*, 2012.

[9] J. Kober, E. Oztop, and J. Peters. Reinforcement Learning to adjust Robot Movements to New Situations. In *Proceedings of the Robotics: Science and Systems Conference (RSS)*, 2010.

[10] J. Peters, K. Mülling, and Y. Altun. Relative Entropy Policy Search. In *Proceedings of the 24th National Conference on Artificial Intelligence (AAAI)*. AAAI Press, 2010.

[11] A. Abdolmaleki, N. Lua, L.P. Reis, J. Peters, and G. Neumann. Contextual Policy Search for Linear and Nonlinear Generalization of a Humanoid Walking Controller. In *Journal of Intelligent and Robotic Systems*, 2016.

[12] A. Abdolmaleki, N. Lua, L.P. Reis, and G. Neumann. Regularized covariance estimation for weighted maximum likelihood policy search methods. In *Proceedings of the International Conference on Humanoid Robots (HUMANOIDS)*, 2015.

[13] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[14] M. Molga and C. Smutnicki. Test Functions for Optimization Needs. In *http://www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf*, 2005.

[15] A. Ijspeert and S. Schaal. Learning Attractor Landscapes for Learning Motor Primitives. In *Advances in Neural Information Processing Systems 15(NIPS)*. 2003.